

Lessons Learned from Weekly Translations of CAD to ArcSDE

Lessons learned from how to perform routine translations of large number of rubber sheeted CAD datasets to ArcSDE 8.X for use with ArcIMS. Overview of how one local government takes 4500 + CAD files with attributes from the city utility and Tax Assessors office to an SDE warehouse for ArcIMS on weekly basis. Solutions of translation issues ranging from annotation to geometry type differences between CAD and SDE. Show how ArcIMS as the end user of data can drive translation decisions. Additionally, an overview of developing a website with tips from our DBA and developer.

Background

KGIS was formed in 1985 to ensure that all mapping efforts within the local government, City of Knoxville, Knox County, and the Knoxville Utilities Board, were done in a consistent manner. KGIS maintains the Knox County land base which consists of planimetry, topography, digital terrain models, and aerial photography. Individual departments maintain data while KGIS provides system administration, database administration, technical support, and GIS application development. Graphic data is stored in MicroStation design files while attribute data is stored in an Oracle database. The data is maintained using a combination of MicroStation, MGE (Intergraph's Modular GIS Environment), and FRAMME (Intergraph's utility mapping software). Users can view KGIS data through a custom built a client-server application that allows read only access to the live GIS data.

In an effort to increase the availability of mapping data to the general public, KGIS decided to create an online GIS mapping site. After looking at several options it was determined that ArcIMS with ArcSDE was the best solution. Therefore, KGIS had to develop extensive data translation routines to move the data between GIS platforms. The translation consists of three types of data:

- Static layers - such as topography and planimetry which are only updated once a year by KGIS.

- Constantly changing layers - such as property and addressing which are updated daily.
- Arbitrarily changing layers - such as school boundaries, city survey control points, and public facilities which change at various intervals.

Data is translated from the production environment (MicroStation, MGE, FRAMME, and Oracle) into an ArcSDE SQL Server 2000 geodatabase. Static layers are translated once a year. Constantly changing layers are translated biweekly. And arbitrarily changing layers are translated on an as needed basis. All data translations are preformed using FME by Safe Software. FME can read all formats (DGN, SDE, ESRI Shapefiles, Oracle, and SQL Server) included in this process and can run unattended.

Development of the website and translation routines was a much larger task than originally anticipated by KGIS. A contractor was hired to assist in development of both the ArcIMS website and data translation routines. After the initial development was completed, KGIS spent approximately six months enhancing the website and translation routines. This paper focuses on the how KGIS overcame several translation issues between MicroStation and ArcSDE. It is important to note that many of the solutions discussed in this paper are only appropriate for snapshot standalone solutions.

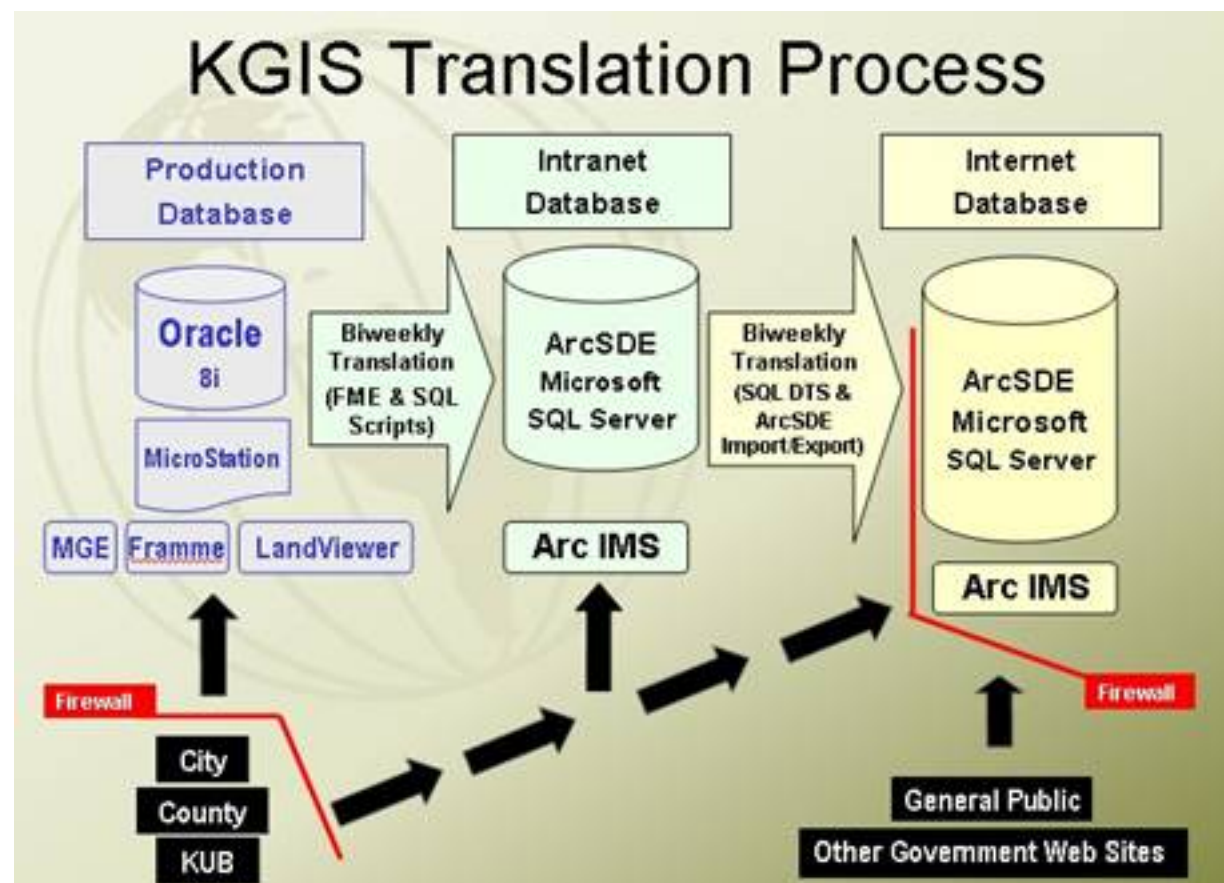


Figure 1 Translation Process

Quick Facts about KGIS and Translated Data

170,000+ parcels translated every two weeks
27,000+ road segments translated every two weeks
228,000 addresses translated every two weeks
Size of ArcSDE database 15 gigabytes
Number of feature classes in ArcSDE 260
Number of Non-utility features inside Intergraph MGE 1126
Number of Utility Features inside Intergraph FRAMME 85
Population of Knox County 382,000+ (2000 Census)
Area of Knox County 526 square miles

Differences between CAD Files and ArcSDE

CAD files are inherently different from GIS data files. GIS data elements are typically broken into three distinct categories - lines, polygons, and points. CAD files have a much larger variety of element types (e.g. lines, polygons, points, curves, ellipses, arcs, single line text, multi-line text, etc.) To further complicate matters, CAD files usually have different element types and different features all stored in the same file. For example: a planimetric file may have buildings as polygons or lines; edge of pavement features drawn as lines, arcs, or ellipses; and tree lines as a pattern element. An ArcSDE feature class on the other hand consists of only one feature (buildings) and one feature type (polygon). Typically, all elements in a GIS file are associated with a database attribute. Again this is not always the case in CAD files. However, CAD files have level, color, weight, and style properties associated with each element that can help distinguish one element from another.

As mentioned earlier KGIS maintains their GIS data in MicroStation CAD files. The elements in the design files are linked to an Oracle database. Most elements have at least one linkage – a feature link – which designates what type of feature the element represents (e.g. building, edge of pavement, greenway, etc.) in the GIS. Some elements have additional linkages for attribution (e.g. address, parcel, or owner information). However, some departments maintain data in CAD files that have little or no GIS data attribution. In these cases the element's level, weight, style, and color can be used to distinguish it from other elements during the translation process. By making use of an element's level, weight, style, and color properties the translation routine can separate it into the correct feature class and in some cases add attribution (an example will be discussed later in this paper).

Translating Text

From a cartographic stand-point, text proved to be one of the most difficult CAD elements to translate. Since text is rendered differently in MicroStation than ArcIMS, translating text elements and keeping them cartographically pleasing, required some planning and lots of experimenting. KGIS decided to break text into three distinct categories:

- **Fixed Text** - Text that needed to retain its cartographic style (appearance and position)
- **Rotated Dynamic Text** - Text that needed to maintain its position and rotation but could vary in size and style
- **Dynamic Text** - Text that had no specific cartographic placement and could vary in size and style (normal label)

For text to maintain its cartographic style, position, and rotation (fixed text) it had to be dropped into line elements before being translated. This allows ArcIMS to render the text elements exactly as they appear in the CAD files. Property dimensions and parcel numbers are examples of features that had to maintain their cartographic positioning and are rendered as individual line elements. Obviously, this solution is not appropriate for maintenance platforms.

Addressing text (house numbers) is stored as a combination point feature and text node in the CAD files. However, house numbers are placed parallel to the road they are associated with. Therefore, it was important that they maintain their rotation as defined in the CAD file. To facilitate this, the translation routine reads the text rotation and creates a line segment at the specified angle. The house number is then added as an attribute to the line segment. The line is then rendered transparent in ArcIMS and the house number is used to label the line. This allows the text to maintain its rotation along the road segments. Originally, the only attribution on the line segment was the house number. However, for identify functions inside ArcIMS the unique identifier for the house number was added. With the unique identifier, the system can use an ASP query to get the additional address information from the address table.

Below is an example of the original CAD file (Figure 2) and an ArcIMS rendered map (Figure 3).

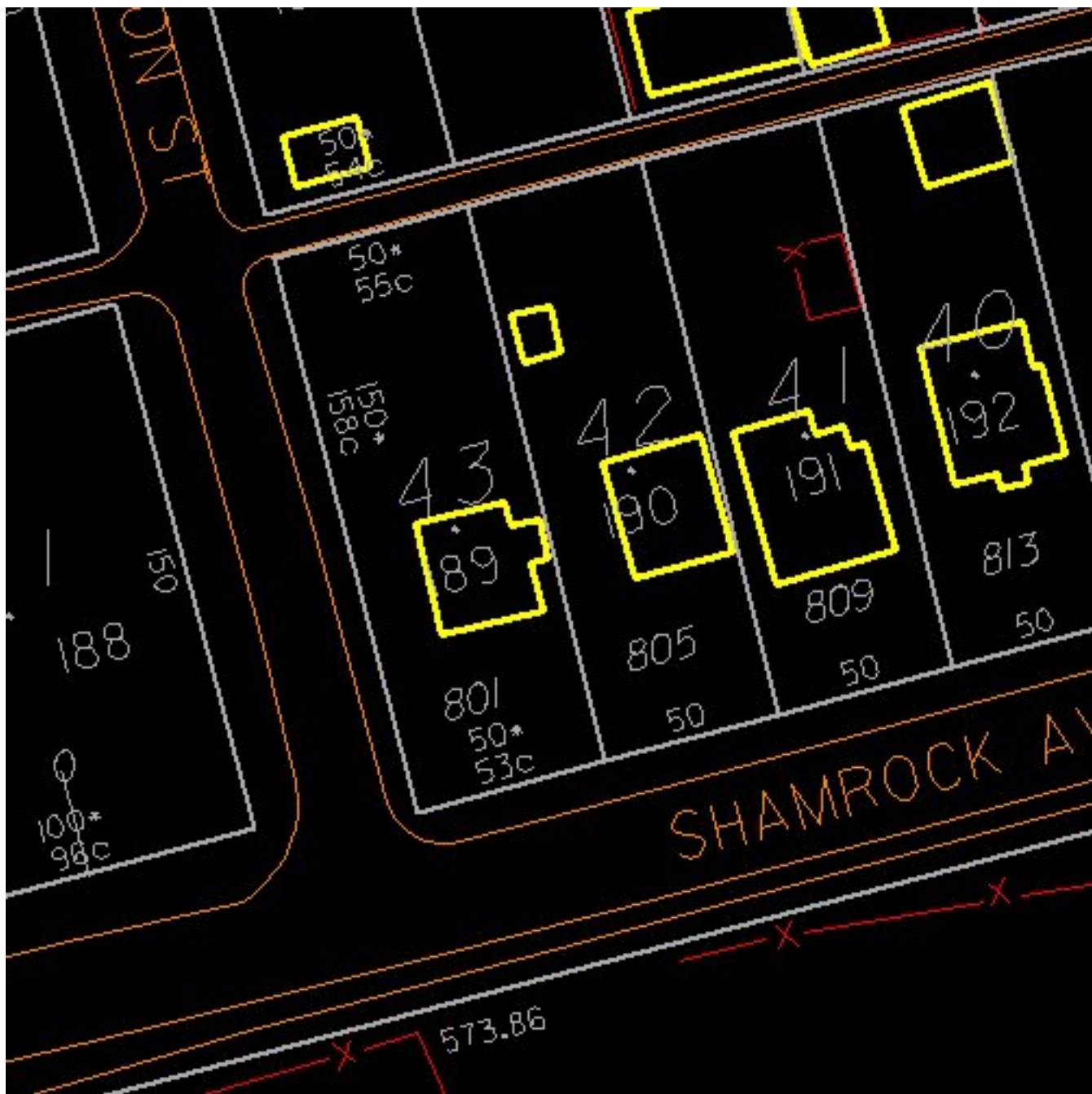


Figure 2 Original CAD File

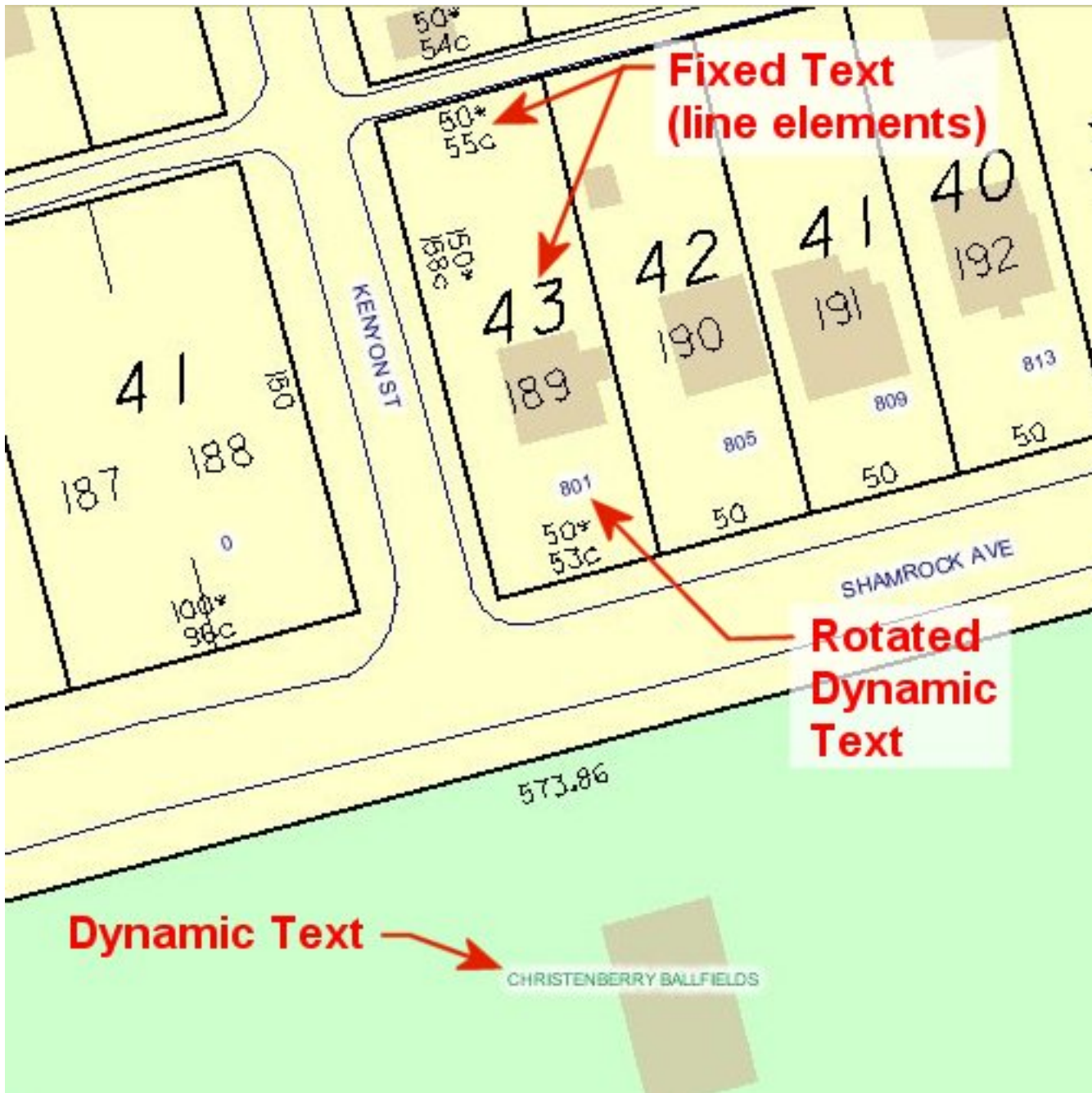


Figure 3 ArcIMS Rendered Map

Translating Cells and Symbols

Cells and symbols from CAD files also create unique translation issues. The local utility company, Knoxville Utilities Board (KUB), maintains data in Intergraph's FRAMME software. Font symbols are used to depict utility features (transformers, switches, valves, poles, etc.). These features must maintain their cartographic positioning (e.g. transformers are tied to a specific line in an electrical schematic). Utility drawings are often complex and cluttered. Labels are placed at specific locations to help indicate which feature they are associated with. In some instances leader lines are used to avoid confusion. The relationship between features and labels

must be maintained.

It was imperative that the utility data maintained its cartographic integrity during the translation. The initial thought was to simply drop the element to line features and translate them into SDE as lines (like the fixed text elements in the above example). However, the user must be able to query and identify the elements in the map. To accomplish this, the translation routine was enhanced to create polygon elements, which represent the utility features, on the fly. The attribution is linked to the polygon features, which allows the user to find, query, and identify the various utility features while keeping the original symbols look and feel. The cells and symbols are then dropped to individual lines and loaded into SDE as simple line features. This also allows the user to see the same symbols between both systems (e.g. transformers look the same on the web or in FRAMME).

Below is an example of a KUB power system schematic. Figure 4 is from CAD file and Figure 5 is from ArcIMS:

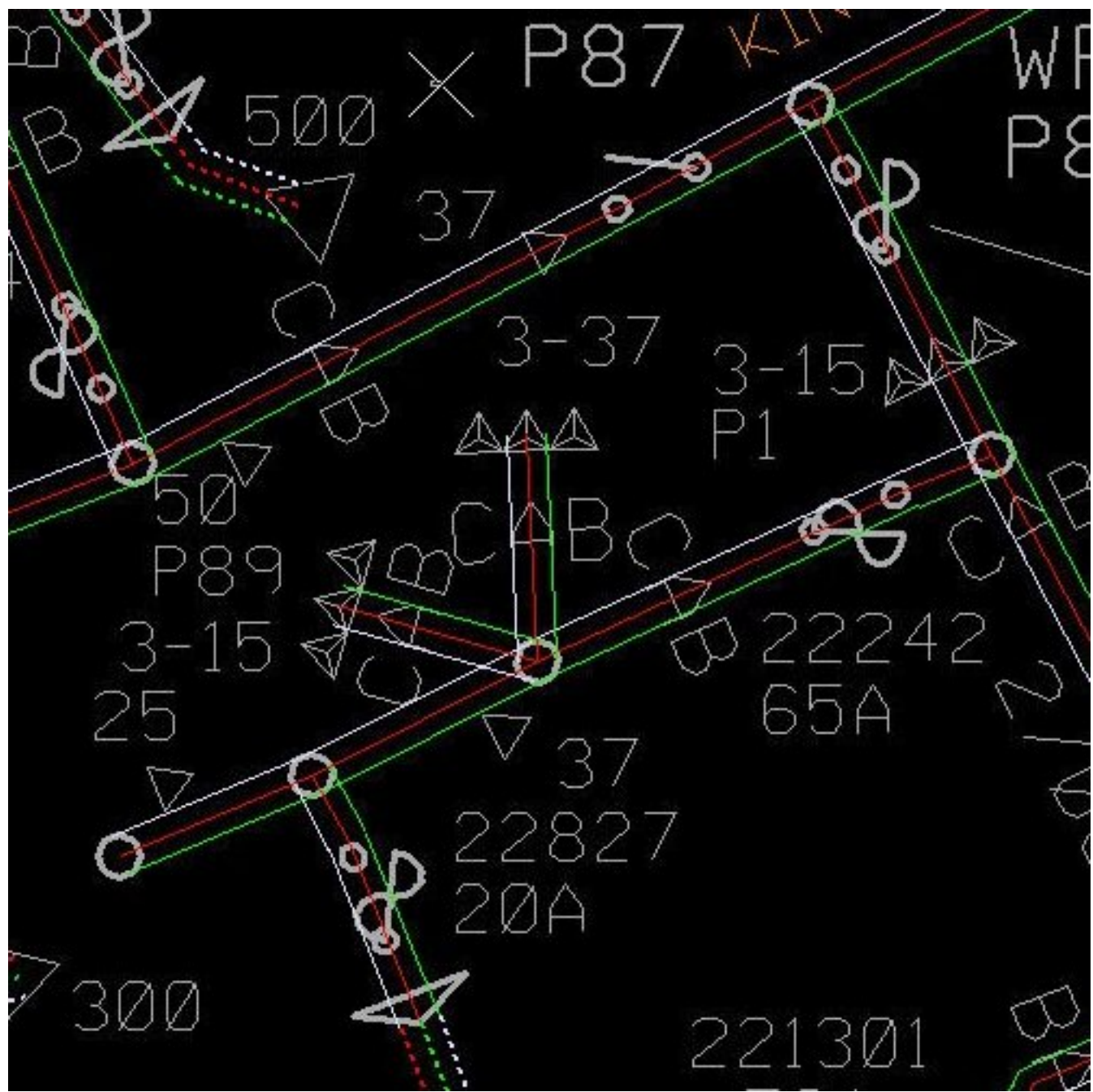


Figure 4 Original FRAMME Electric Data

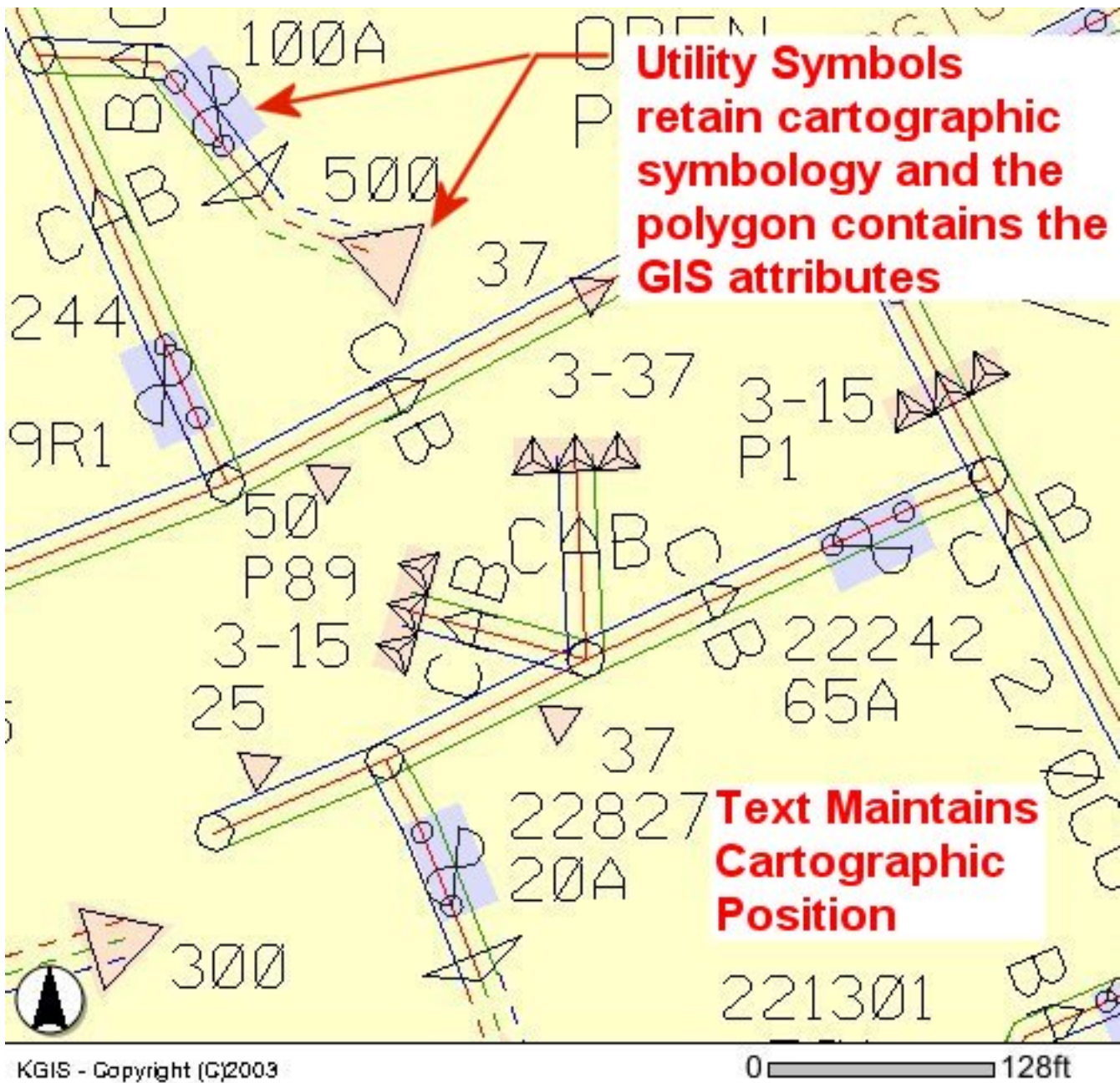


Figure 5 ArcIMS Electric Data

Enhancing Data during the Translation

It is also possible to enhance data during the translation process. Obviously the utility example above shows a great way to enhance data by creating new features during the translation. However, FME also has some additional built in methods for enhancing the data.

Often times CAD files do not have attribution attached to the elements. FME allows you to do some spatial intersections during the translation process. KGIS recently used this method to

attribute city blocks and ward boundary polygons with text data from the CAD file. The CAD file contained polygons for the city blocks and ward boundaries. It also contained a label in the center of each polygon for the city block number and ward number. During the translation, the insertion point of the text was overlaid with the polygons and the polygons attributed with the value of the text string. Obviously, GIS software packages offer utilities to analyze the spatial relation between the text and polygons, but FME can perform this step during translation saving you a step.

Donuts and Multi-Part Polygons

KGIS has parcels that are multi-part polygons, donuts, and/or islands. Donut polygons present an interesting problem because MicroStation treats the parcel as two individual polygons. The outer polygon is flagged as a “solid” while the inner polygon is flagged as a “hole”. ArcSDE on the other hand, requires vertices on inner polygon to be drawn in the opposite direction as the vertices on the outer polygon. FME overcame both these issues by grouping the polygons by the parcel id which creates a multi-part polygon. While using the inner (hole) polygon to cut the outer polygon, FME is able to re-orient the vertices to ensure that the polygons translate into ArcSDE.

Re-orienting polygon vertices in FME was the most recent improvement to the parcel translation. Previously, an ArcScript called “CleanShapefile” was used to re-orient the vertices. In this process the donut geometry was first translated to a shapefile and then into ArcSDE. Shapefiles were used because they do not test donut geometry as rigorously as ArcSDE. A common donut polygon in the KGIS dataset is a condo complex with joint ownership of common ground shown below:



Figure 6: Parcel Consisting of a Complex Donut Polygon

Optimize for Web Performance

ArcIMS allows the user to use value-based rendering which renders each element differently based on an attribute in the database. This is a great feature but requires every element in the layer that is within the view extent to be checked. This is a labor intensive process and should be used carefully on large datasets. Value based rendering works well on relatively small datasets like school districts, political districts, or utility districts. KGIS found that splitting a feature class by rendering attributes to create smaller datasets greatly increased performance.

Roads are a great example from the KGIS dataset where value-rendering was enhanced by creating smaller datasets. KGIS translated all the roads into one layer and used a combination of scale-based and value-based rendering on over 27,000 road segments. The only roads shown at full extent (County wide map) were interstates, arterials, and collectors but the map generation time for this map was unacceptable. Interstates, arterials, and collectors account for less than 30% of all road segments – local roads make-up the other 70%. To simplify the map generation process, the centerline feature class was split into an additional 4 feature classes – Interstates, Arterials, Collectors, and Local Roads. ArcIMS now renders each feature class independently and no longer has to check the transportation class of each road segment before it rendered. Our map generation time on a county wide map dropped by over 70%.

Another consideration for web performance is the database scheme. The database scheme used on the production database is great from maintaining data but can be cumbersome for doing simple lookups for web pages. For example: due to character limitations in the property assessor's software, owner information is split across 4 database columns. Additionally addresses are stored as house number, unit number, street name base, street name type, etc. With some simple SQL scripts this data can be combined into new columns and used throughout the web pages. This simplifies queries and in some cases actually improves performance.

Again these solutions work because the database is a snapshot of the production database are not recommended for data maintenance systems.

SDE Storage (SQL Server)

KGIS stores the data under separate databases for security and administration ease. KGIS created users that were read only for each database or read only users that can read combinations of databases. KGIS has a utility reader that can see all databases and a non-utility reader that sees everything except utility data.

For security and backup/restore purposes, KGIS placed the utility data in its own database (note that in SQL server many databases make up a single instance). KGIS also separated static layers (e.g. topology and planimetry) from non-static layers (e.g. parcels and addresses), since land base data is updated annually and parcels are updated bi-weekly. This makes database backup/recovery easier since parcels, addresses and other changing layers only account for 30% of the ArcSDE data.

The “Adds” and “Deletes” tables within ArcSDE store changes to a layer if edited via ArcMap. Use caution when working with ASPs against tables that have been edited within ArcMap. KGIS noticed a problem when park names were edited via ArcMap. The map rendered the new park names but a list box in the ASP still displayed the original names. The database must be

compressed to incorporate the “Adds” and “Deletes” tables. To edit geometry without compressing database – export the feature to a shapefile, make the edits, and import back to ArcSDE. This is not a major issue but Developers need to be aware of any edits to the data to ensure database compression is performed.

Shapefiles can be used with ArcIMS, but if SDE is available it is easier to keep all data in that single system. Also shapefiles are locked while in use. Developers can add or modify attributes within ArcSDE database without system interruption. ASP Lookup pages can interact with databases in more complex methods than shapefiles, creating a more useful website. Large shape files can be slow to render relative to ArcSDE due to inherent speed advantages gained from indexes within enterprise databases.

ArcSDE vs. CAD File Precision

Precision and scale are beyond the scope of this paper. However, they are important and deserve a quick note. ArcSDE and MicroStation have similar scale and precision. To calculate the minimum precision needed in ArcSDE use the calculation below. For example: KGIS uses MicroStation with the following settings:

- Master Units – Survey Feet
- Sub Units – Inches
- Sub Units per Master Units = 12
- Positional Units per Sub Unit = 100

Using the KGIS settings shown above the minimum precision needed to ensure polygons do not self-intersect would be:

(sub units per master unit) x (positional units per sub unit)

$$12 \times 100 = 1200$$

An example that highlights the need for understanding precision is show below.

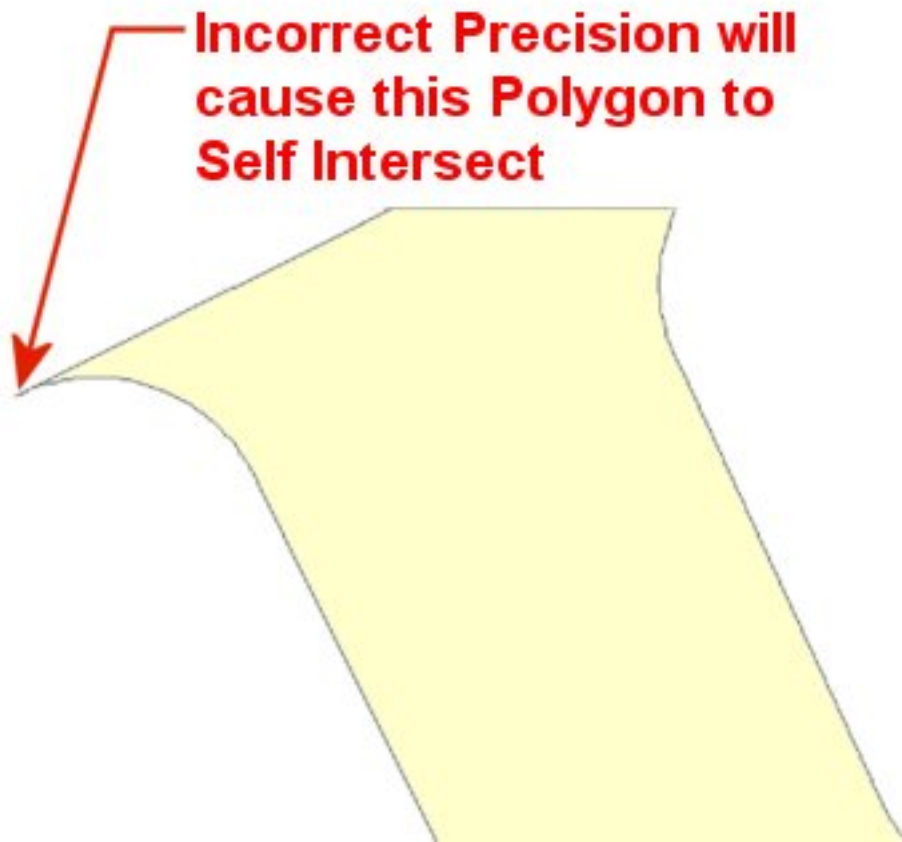


Figure 7 Precision Polygon

CONCLUSION

Translating data can be an overwhelming task, but using third party translations tools and understanding the end use of data, will ensure a successful translation. For web warehousing, translation should be completed prior to web page development to avoid continuous editing of lookup pages and AXL files. Translations can then be refined as web development progresses. Annotation will be the most complicated to translate, but dropping text to lines will ensure accurate rendering. The key to translation is producing a final product that looks as good as or better than the original CAD data while maintaining all the GIS attributes.

REFERENCES OR ACKNOWLEDGMENTS

Michael Baker and Assoc., Virginia Beach Office for initial development of website and initial

set of FME translation files.

Bryan Lynn, Programmer, KGIS
Conn McAnally, DBA, KGIS